42 | 如何整理出我们自己的可视化工具集?

2020-10-14 月影

跟月影学可视化

进入课程 >

讲述：月影

时长 12:01　大小 11.02M

你好，我是月影。

我们知道，可视化虽然可以认为是前端的一个小分支，但总体来说它也是一个比较有广度的领域，从内容上大体包括数据图表、图形绘制、地理信息可视化、3D 场景和数字孪生应用等等，从设备上包括移动端、PC 浏览器和可视化大屏等等。

作为刚刚入门可视化领域的工程师，我们经常会因为知识面不够广、工具不够丰富，而寄望于用单一的工具来解决我们遇到的所有问题，所谓“手里有锤子，看什么都像是钉子”。想想也知道，这并不能很好地处理所有的可视化应用。因为，一旦我们选择了错的、不适合于当前场景的工具，就会让最终实现的结果不理想，或者开发效率不高。

因此，想要成为一名优秀的可视化工程师，我们必须要知道如何丰富自己手中的工具集，来应对各种场景和挑战。丰富工具集，并不意味着我们需要从头开发一切适合各种场景使用的工具。大部分时候，我们可以利用行业发展和社区的力量，站在巨人的肩膀上，使用各种成熟的工具，来更好地完成我们的项目。

那今天，我带你一起系统地梳理目前行业里，适合可视化应用的优秀且成熟的工具，来丰富你的工具箱，让你在面对各种可视化需求时能够游刃有余。

首先，我们来看看可视化应用的主体需求分类，搞清楚了这些，我们才能选择出更合适的工具来完成这些需求。根据可视化场景，主体需求一般可以分成 5 种，分别是绘制基本图形、绘制基础图表、绘制关系图和流程图、绘制地理信息以及绘制三维模型和数字孪生应用。接下来，我就一一来说。

主体需求	示例
绘制基本图形	根据数据绘制一些二维、三维的几何图形，它们不一定是完整的图表，通常是具有灵活性和视觉冲击力的小图形，例如粒子效果
绘制基础图表	通常是指绘制常见的饼图、折线图、柱状图、面积图、雷达图、漏斗图等等
绘制关系图和流程图	力导向图、树状图、流程图、脑图等等
绘制地理信息	结合地图、3D地球，对地域相关的信息进行可视化
绘制三维模型和数字孪生应用	三维建筑、人物模型，模拟整个城市、地区的真实场景

极客时间

选择绘制基本图形的工具

我们先来说说绘制基本图形要选择的工具。简单情况下，我们用浏览器原生的四类基本图形系统就可以完成绘制，分别是 HTML/CSS、SVG、Canvas 和 WebGL。

HTML/CSS 比较简单，也有很多丰富的前端领域的工具，这里就不多说了。SVG 原生的 API 使用起来也并不复杂，不过我们也可以用一些成熟的绘图库，它能够让我们更方便地绘制各种图形。我比较推荐 Snap.svg 这个库，它提供的 API 能够非常方便地绘制各种

SVG 图形。尤其值得赞扬的是它提供了一套交互式教学文档，能够让你快速上手整个库的使用。

如果你使用 Canvas 和 WebGL 来绘制图形，可选的工具就比较多了。比较成熟的 2D 渲染引擎包括 Fabric.js、Pixi.js 等，比较成熟的 2D/3D 引擎包括 SpriteJS、P5.js 等，以及比较成熟的 3D 引擎包括 ThreeJS、Babylon.js 等。这些库各有特点，它们的官方使用文档和示例都比较完善，我建议你多花一些时间去学习它们的使用方法。这能帮助你在绘制基本图形的时候，灵活选用适合的库。

另外，既然说到了 Canvas2D 绘制图形，那我还要提一下Rough.js这个库。这是个非常有特点的库，它能够绘制出带有手绘风格的图形，比如，我们在第 5 课就使用过它。

我刚才说的这些工具，对于绘制基本的 2D/3D 图形来说已经足够了。但是，功能相对更强大和丰富的库，体积也会更大一些，使用上也会复杂一些，所以我们还是应该根据实际需求复杂程度来灵活选择。

选择绘制基础图表的工具

绘制基础图表是可视化项目中很常见的需求，我们可以采用图表库和数据驱动框架来完成。常用的成熟图表库包括 ECharts、Chart.js、Highcharts、AntV G2 这 4 种。大部分图表库在使用上大同小异，效果也差别不大，好像选什么都没有差别。但在选择图表库的时候，我们也需要考虑底层的图形系统或图形库，这关系到复杂图表渲染和交互的性能，另外在同时需要绘制图表和基本图形的时候，选择统一的图形系统可以保持一致性，也能更好地实现图表与图形的协同交互。

这里，我就说几个常见的搭配组合。如果你使用 SpriteJS 作为底层图形库，还可以选择 QCharts。如果你是在移动端设备渲染图表，可以考虑使用AntV F2，这是一个专为移动端场景设计的图表库。如果你要绘制更加灵活的图表，以及关系图和流程图，可以选择数据驱动框架，例如D3.js。不过 D3.js 虽然很好用，但它是一个足够复杂的框架，如果你希望在可视化领域深入发展，最好能再多花一些时间，彻底掌握它的使用方法。

最后，我还想再说说Vega这个库。这也是一个图表库，它定义了一套基于 JSON 规范的可视化语法，以声明式的方式来绘制各种图表。最关键的是，Vega 定义可视化语法规范的

思路，对我们自己设计和实现图表库有着非常大的借鉴意义。如果你打算自己设计一套图表库，我希望你能好好研究一下 Vega，相信你会有所收获的。

选择绘制关系图和流程图的工具

接着，我们再来说一类特殊的图表，比如关系图、流程图、脑图等等。我们一般将它们单独归为一类应用，称为图可视化。

图可视化怎么实现呢？我们在第 38 课里使用 D3.js 实现过相关的例子。除了 D3.js 以外，还有一类直接绘制这些图形的图可视化库，常用的有Mermaid.js、Sigma.js以及AntV G6等等。

其中 Mermaid.js 量级更轻，主要是以声明的方式来绘制各种流程图。而 Sigma.js 和 AntV G6 的功能更丰富，实现的图可视化不仅类型更多，还能包含复杂的用户交互效果。

此外，还有一个特殊的库Dagre。它是绘制流程图的底层库，主要是用来计算图的元素布局，使用它再结合图形库，我们就能自己实现一个绘制流程图的图可视化库。这个图可视化库实现起来也不是很难，我把这个任务留在本节课的末尾，你可以试着去挑战一下。

选择地理信息可视化工具

前面两节课，我们一起完成了 3D 地理信息可视化的实战。地理信息可视化是可视化项目中很重要的一类应用，大部分可视化大屏应用场景都会包含地图或者 3D 地球，因为许多数据本就和地理信息相关。由于需求很多，因此市场上有许多地理信息可视化的库可供我们选择。

比较成熟的地理信息可视化库包括MapBox、MapTalks、Leaflet.js、MapV、AntV L7等等，它们都支持简单的 geoJSON、topoJSON 数据和分片加载的瓦片地图。另外 d3 的子模块d3-geo也能够处理地理信息可视化，尤其是它提供了多种地图投影，非常适合与其他库联动，实现各种不同的地图场景应用。

总体而言，地理信息可视化是可视化领域里比较复杂的方向，这些地理信息可视化库的功能也较为丰富，使用场景很多，如果详细来讲的话，内容足够支撑起一门单独的课程了。我们前面花了 3 节课来讲的地理信息可视化案例，其实也只是给地理信息可视化应用开了

一个头，如果你有兴趣深入学习，可以通过这些地理信息可视化库，再结合实战来真正深入掌握地理信息可视化的方法，这也是成为优秀可视化工程师的必经之路。

处理三维模型和数字孪生应用

在学习 3D 图形的绘制的时候，我讲到，3D 绘制一般有两种方式，一种是加载静态的 3D 模型数据，然后将 3D 物体渲染出来。这些 3D 模型数据，通常是通过设计工具离线生成的。这种应用场景在游戏领域比较常见。

SpriteJS加载的3D模型数据，图片来源：spritejs.org

另一类则是动态加载 3D 几何模型，用前面绘制基本图形的工具就可以实现。在可视化应用中，这类场景通常更普遍。

动态加载的几何图形，图片来源：OGL

不过，在可视化领域中，有一类应用也会用到非常多的 3D 模型，那就是数字孪生应用。所谓数字孪生，是对物理世界实体或系统的数字化表达。简单来说，就是在虚拟世界中，通过 3D 渲染来还原真实物理世界，这需要我们将现实世界中的物体模型化为虚拟世界中的 3D 几何体。

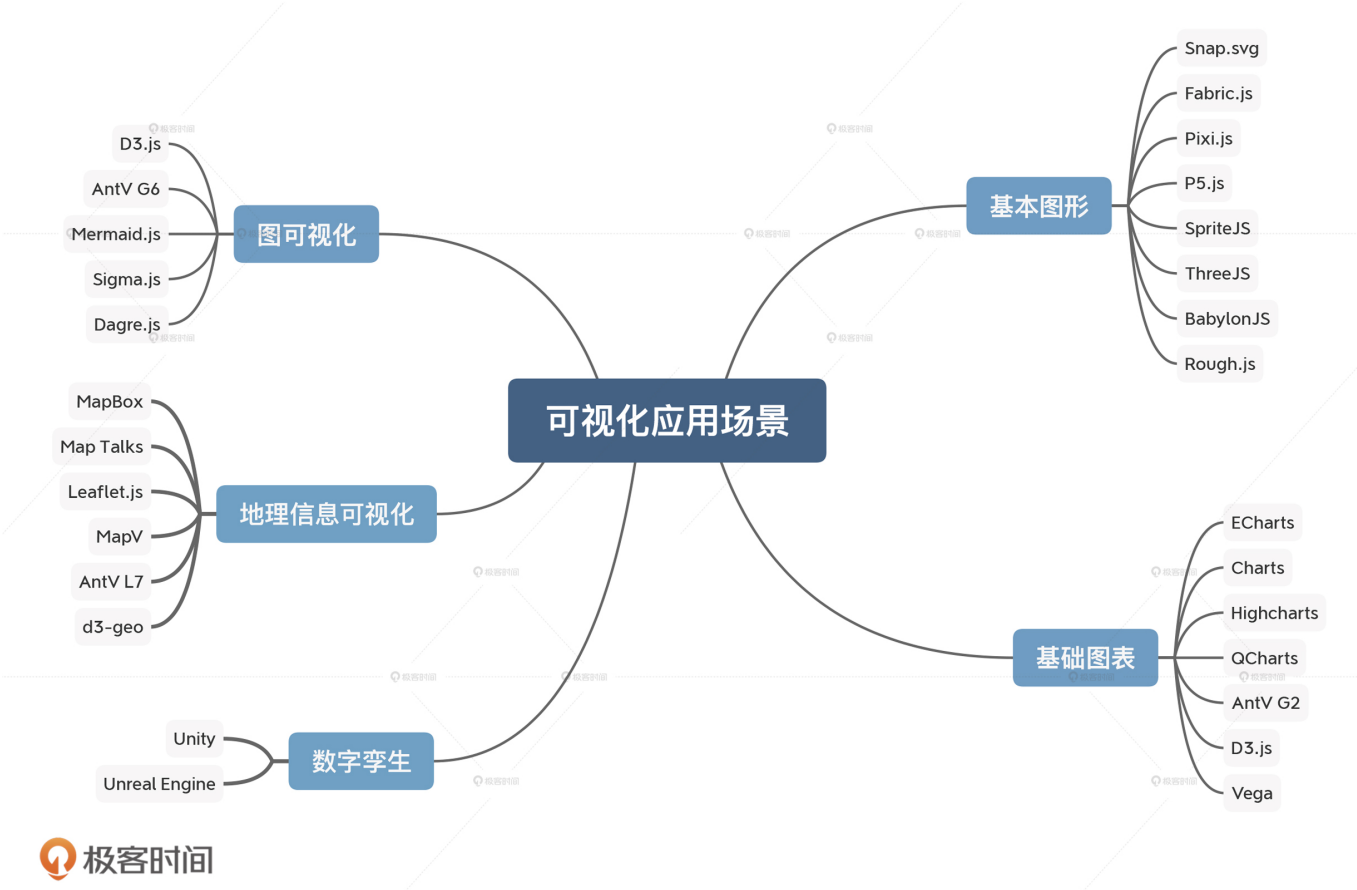
物理世界和数字孪生示意图，图片来源：36kr.com

在这样的应用场景中，有时候我们可以考虑采用游戏的方式，使用游戏引擎和框架，例如Unity或者虚幻引擎来完成我们的可视化应用。当然，这也就进入了另一个领域，游戏创作的领域。这部分内容，我们简单了解一下就可以了。

要点总结

可视化有着丰富的使用场景，大致上可以分为 5 类，分别是：绘制基本图形、绘制基础图表、图可视化、地理信息可视化以及数字孪生。

这些场景各自有适合的工具库和框架可供我们选择，我在这里整理了一个脑图，供你在具体应用中参考。



这是我们专栏最后一个模块的最后一节课了，在这几个月的时间里，我们先后学习了图形学与数学、视觉基础、视觉高级以及数据处理的相关知识。其中，我们通过一些可视化实战，学习了部分图形系统与图形库和工具的使用，这当中主要包括 SpriteJS、D3.js、QCharts 等等。这和我们这节课列出的各类工具相比，可以说是非常少了。

我想有的同学可能会有疑问，还有这么多的工具我们并没有详细来讲，那我学习到这里，算是真正入门可视化了吗？我想说的是，工具和库都是为我们服务的，正确使用这些工具

和库的基础，其实就是我们前面学过的图形学与数学基础、视觉呈现技术，以及数据处理的原则，这些内容才是可视化的本质。而上层的工具和库的使用虽然复杂，但并不难，基本上照着文档学习和实践，就能够一步步掌握。

而如果缺少扎实的基础，那么在使用工具遇到缺失功能或者性能问题的时候，你就可能会束手无策。因此，这门课，更多的还是为你打好基础，领你进入可视化的世界，今后的路，就要靠你自己一步一步地走了。

小试牛刀

今天，我特意提到了 Dagre 这个库。它是一个基础库，经常用来给流程图的元素布局，不涉及渲染的部分。你能用它作为基础，再以 SpriteJS 为渲染引擎，来实现一个高效、强大的流程图库吗（提示：在 API 方面你可以借鉴 Mermaid.js）？

除此之外，这节课，我们介绍的工具和库非常多，你肯定很难在短时间内完全掌握，但我也不希望你只是泛泛了解。所以，我希望你能选择自己最感兴趣的同类库中的两个库，利用我之前在 GitHub 仓库里留的北京天气和空气质量数据，分别使用它们做一些可视化应用，比较两个库在使用上的不同，以及各自的优、劣势。

欢迎把你实现的效果分享到留言区，也欢迎你把自己在工具整理上的心得体会分享出来，我们一起交流！

推荐阅读

[1] Snap.svg、[2] Fabric.js、[3] Pixi.js

[4] SpriteJS、[5] P5.js、[6] ThreeJS

[7] Babylon.js、[8] Rough.js、[9] ECharts

[10] Chart.js、[11] Highcharts、[12] AntV G2

[13] QCharts、[14] AntV F2、[15] Vega

[16] Mermaid.js、[17] Sigma.js、[18] AntV G6

[19] Dagre、[20] MapBox、[21] MapTalks

[22] Leafletjs、[23] MapV、[24] AntV L7

[25] d3-geo、[26] Unity、[27] Unreal Engine

提建议

© 版权归极客邦科技所有，未经许可不得传播售卖。 页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 41 | 实战（五）：如何实现3D地球可视化（下）？

下一篇 国庆策划 | 假期别闲着，一起来挑战“手势密码”

精选留言

写留言

> 由作者筛选后的优质留言将会公开显示，欢迎踊跃留言。